

**What is claimed is:**

1           1. A method for accessing a unified memory in a micro-processing system having  
2 a microprocessor, a one level pipeline, and a two-phase clock, such that all instructions  
3 are executed in a single cycle, comprising:

4           (a) fetching a program instruction from the unified memory;

5           (b) determining if the fetched program instruction would require three unified  
6 memory accesses during a single instruction cycle for proper execution of the fetched  
7 program instruction, proper execution of the fetched program instruction being the  
8 microprocessor performing the operations requested by the fetched program instruction in  
9 a single instruction cycle;

10          (c) accessing the unified memory a first time, during the instruction cycle  
11 associated with the fetched program instruction, with a dummy access when it is  
12 determined that the fetched program instruction requires three unified memory accesses  
13 for proper execution of the fetched program instruction;

14          (d) fetching a next program instruction from an instruction register, during the  
15 instruction cycle associated with the fetched program instruction, when it is determined  
16 that the fetched program instruction requires three unified memory accesses for proper  
17 execution of the fetched program instruction; and

18          (e) accessing the unified memory a second time, during the instruction cycle  
19 associated with the fetched program instruction, with a data access when it is determined  
20 that the fetched program instruction requires three unified memory accesses for proper  
21 execution of the fetched program instruction.

1           2. The method as claimed in claim 1, wherein the data access is a read data  
2 access.

1           3. The method as claimed in claim 1, wherein the data access is a write data  
2 access.

1           4. The method as claimed in claim 1, wherein the fetched program instruction is a  
2 last instruction of a loop.

1           5. The method as claimed in claim 1, wherein the instruction register is an  
2 instruction stack, thereby enabling program instruction fetches for nested loops.

1           6. A method for accessing a unified memory in a micro-processing system having  
2 a microprocessor, a one level pipeline, and a two-phase clock, such that all instructions  
3 are executed in a single cycle, comprising:

4           (a) fetching a program instruction from the unified memory during a first  
5 instruction cycle;

6           (b) determining if the fetched program instruction for a second instruction cycle is  
7 a conditional program code discontinuity;

8           (c) accessing the unified memory a first time during the second instruction cycle  
9 with a dummy access when it is determined that the program instruction accessed for a  
10 second instruction cycle is a conditional program code discontinuity; and

11           (d) accessing the unified memory a second time during the second instruction  
12 cycle to read a new instruction when it is determined the program instruction accessed for  
13 a second instruction cycle is a conditional program code discontinuity, thereby delaying  
14 the instruction access from the unified memory for the second instruction cycle by a half  
15 cycle.

1           7. The method as claimed in claim 6, wherein the conditional program code  
2 discontinuity is a jump instruction.

1           8. The method as claimed in claim 6, wherein the conditional program code  
2 discontinuity is a call instruction.

1           9. A method for accessing a unified memory in a micro-processing system having  
2 a microprocessor, a one level pipeline, and a two-phase clock, such that all instructions  
3 are executed in a single cycle, comprising:

4 (a) fetching a program instruction from the unified memory;  
5 (b) determining if the fetched program instruction is a loop initiation instruction;  
6 (c) storing a first instruction of the loop in an instruction register when the fetched  
7 program instruction is a loop initiation instruction;  
8 (d) executing the loop;  
9 (e) determining if a fetched instruction during the execution of the loop is a last  
10 instruction of the loop;  
11 (f) accessing the unified memory a first time, during the instruction cycle  
12 associated with the fetched last instruction of loop, with a dummy access;  
13 (d) fetching the first instruction of the loop from the instruction register, during  
14 the instruction cycle associated with the fetched last instruction of loop; and  
15 (e) accessing the unified memory a second time, during the instruction cycle  
16 associated with the fetched last instruction of loop, with a data access.

1 10. The method as claimed in claim 9, wherein the data access is a read data  
2 access.

1 12. The method as claimed in claim 9, wherein the data access is a write data  
2 access.

1 13. The method as claimed in claim 9, wherein the instruction register is an  
2 instruction stack, thereby enabling program instruction fetches for nested loops.

1 14. A method for accessing a unified memory in a micro-processing system  
2 during a loop instruction, comprising:

3 (a) accessing a program instruction from the unified memory during a first  
4 instruction cycle;  
5 (b) determining a type of program instruction;  
6 (c) pre-fetching a next instruction from the unified memory;  
7 (d) saving the pre-fetched instruction in a register when it is determined that the  
8 type of program instruction is a first instruction of a loop;

9           (e) fetching a next instruction from the register when it is determined that the type  
10 of program instruction is a last instruction of a loop;  
11           (f) accessing the unified memory with a dummy access during execution of the  
12 last instruction of the loop; and  
13           (g) accessing the unified memory, a second time, with a data access during  
14 execution of the last instruction of the loop.

1           15. The method as claimed in claim 14, wherein the method saves the pre-fetched  
2 instruction in a stack register when it is determined that the type of program instruction is  
3 first instruction of a loop to enable nested loops and interruptible loops, and the method  
4 fetches a next instruction from the stack register when it is determined that the type of  
5 program instruction is a last instruction of the loop.